Guidance of Reusable Launchers: Improving Descent and Landing Performance

Pedro Simplício* and Andrés Marcos†
University of Bristol, BS8 1TR Bristol, United Kingdom

Samir Bennani‡
European Space Agency, 2201 AZ Noordwijk, The Netherlands

The sizing and capability definition of reusable launchers during high-speed recovery are very challenging problems. In this article, a convex optimisation guidance algorithm for this type of systems is proposed based on performance improvements arising from the study of the coupled flight mechanics, guidance and control problem. In order to appreciate the obtained improvements, trade-off analyses of powered descent and landing scenarios are presented first using traditional guidance techniques. Subsequently, these results are refined by using the proposed online successive convex optimisation-based guidance strategy. The *DESCENDO* (Descending over Extended Envelopes using Successive Convexification-based Optimisation) algorithm has been designed as a middle-ground between efficiency and optimality. This approach contrasts with previous convexification algorithms that either aimed at increasing computational efficiency (by typically disregarding aerodynamic deceleration) or reaching trajectory design optimality (by using exhaustive convex approximations). More critically, the algorithm is not confined to the mild coverage conditions assumed by previous approaches and can successfully handle the incorporation of the operational dynamics of reusable launchers. Insights provided by *DESCENDO* operating in a closed-loop fashion over full recovery scenarios enable a computationally-efficient mission performance assessment.

I. Introduction

Space descent and landing (D&L) has its roots in the Apollo program [1] and has since been successfully applied in a variety of missions, ranging from the exploration of Mars [2] and other planets to missions targeting smaller bodies such as JAXA's Hayabusa [3] and Hayabusa-2 (which successfully delivered two landers to asteroid Ryugu in September 2018), ESA's Rosetta [4] and NASA's OSIRIS-REx [5] (which is currently on its way to asteroid Bennu). But more recently, an additional interest on powered D&L on Earth has been growing following the technical proof of

*PhD Candidate, Aerospace Engineering Department, University Walk.
†Senior Lecturer, Aerospace Engineering Department, University Walk. Senior Member AIAA.
‡GNC Launcher Systems Expert, ESTEC GNC Section, Keplerlaan 1. Senior Member AIAA

launcher reusability at commercial level by companies like SpaceX and Blue Origin. Recovery and reuse of launcher stages has the benefit of reducing launch costs while increasing mission responsiveness [6].

Pinpoint landing on and launching from Earth is particularly challenging due to the high levels of aerodynamic and thermal stress induced by its dense atmosphere on the vehicle's structure, which are further worsened by the existence of uncertain and time-varying wind perturbations. These effects become even more adverse for the next-generation of launchers, as lighter (and more flexible) structures tend to lead to stronger control-induced interactions [7].

From a guidance and control (G&C) point of view, this challenging problem only became feasible in the past decade thanks to the increase of computational power available onboard, which enabled a paradigm shift known as *Computational G&C* [8]. This shift was supplemented by additional mathematical developments in the domain of convex optimisation [9–12] among others. Such an approach has been demonstrated with Masten Space's vertical take-off and landing platform Xombie [13–16], which uses a vision system to determine its location and an algorithm termed guidance for fuel-optimal large diverts (G-FOLD) to optimally fly to the landing site.

In addition to the advanced algorithms just mentioned, much knowledge can be obtained (and then leveraged for improvements) from considering space D&L approaches such as constrained terminal velocity (CTV) guidance [17–20]. The main strength of this approach lies in its simplicity and in the fact that, as the main parameters on which it depends are intrinsically representative of the physics of the recovery problem, it allows to have a quick understanding of the acceptable flight performance. This insight is particularly useful since addressing competing requirements such as propellant consumption and aerothermal loads in a combined way is extremely challenging [21–23].

The work in this article is part of an ESA-sponsored activity focused on the management of competing requirements via advanced closed-loop guidance algorithms and robust attitude control, including active load prediction and relief capabilities. Under the same activity, a reusable launcher benchmark has been developed [24] to enable a thorough understanding of reusable flight mechanics as well as the fundamental interactions with G&C algorithms and mechanisms. This benchmark simulates the launch and recovery of a vertical take-off and landing booster used as first stage of a lightweight, non-winged vehicle. Also in that reference, a baseline recovery algorithm based on CTV guidance is proposed and employed for preliminary performance assessment.

This article aims to exploit the existing room for guidance improvement shown in [24], with special emphasis on the aforementioned build-up of knowledge from simpler D&L approaches towards a more sophisticated algorithm coined *DESCENDO* (Descending over Extended Envelopes using Successive Convexification-based Optimisation). In particular, the CTV technique is used to develop trade-off maps of key performance metrics, which are then improved by *DESCENDO*. This algorithm follows similar lines to [25, 26], but it is specifically tailored to the extended flight envelope encountered by reusable launchers. Furthermore, the *DESCENDO* algorithm is implemented in a closed-loop fashion (in opposition to an offline setting) and verified using complete reusable launcher recovery scenarios which, to the best of the authors' knowledge, has not yet been formally investigated.

The layout of the article is as follows. Section II begins with a brief description of the reusable launcher model and recovery missions profiles. Then, in Sec. III, a flight mechanics analysis is performed, which drives the first improvements to the guidance approach. Subsequently, the full *DESCENDO* algorithm is developed and detailed in Sec. IV, and the results of its application to downrange landing and return to launch site missions are presented in Sec. V. Finally, the conclusions are provided in Sec. VI.

II. Reusable Launcher Benchmark

This article relies on the nonlinear 6 degrees-of-freedom reusable launch vehicle (RLV) developed in [24] to study the critical coupling between reusable flight mechanics, guidance and control. It simulates the launch and recovery of a mixed-fuel vertical take-off and landing booster used as first stage of a lightweight, non-winged vehicle launched from the European Space Centre in French Guiana to inject a 1,100 kg satellite in a quasi-polar orbit at 800 km.

The vehicle is mainly steered via thrust vector control (TVC), but two pairs of fins are included to provide attitude control under low thrust and two pairs of cold gas thrusters for low dynamic pressure conditions. The benchmark also includes a baseline closed-loop guidance law for retro-propulsive entry, descent and pinpoint landing, as well as standard algorithms for attitude control computation and allocation.

Equations of motion are written using Earth-Centred Inertial (ECI) [27] and body-fixed frames (see Fig. 1). They are based on the initial states $\{\mathbf{r}_\mathrm{I}(0),\ \mathbf{v}_\mathrm{I}(0),\ \mathbf{q}_\mathrm{B}^\mathrm{I}(0),\ \omega_\mathrm{B}(0)\}$ and on the assumption that effects related to moving masses (including "tail-wags-dog" moment and rocket jet damping) are negligible for trajectory assessment. Mass variability is implicitly accounted for by the mass-depletion dynamics, as it will be seen in Eq. (5).

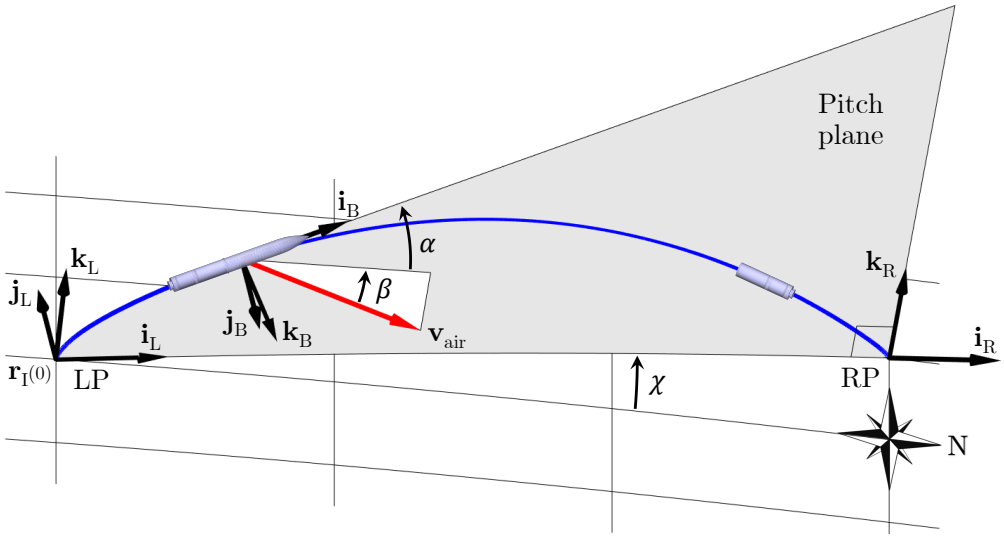


Fig. 1 Relationships between local (launch and recovery pad) and body-fixed reference frames

The vehicle's translational motion (i.e. acceleration $\ddot{\mathbf{r}}_{\mathrm{I}}(t)$ and velocity $\mathbf{v}_{\mathrm{I}}(t)$) is described in the ECI frame by:

$$\ddot{\mathbf{r}}_{\mathrm{I}}(t) = \dot{\mathbf{v}}_{\mathrm{I}}(t) = \mathbf{g}_{\mathrm{I}}(t) + \frac{1}{m(t)} \left[\mathbf{F}_{\mathrm{aero,I}}(t) + \mathbf{F}_{\mathrm{TVC,I}}(t) + \mathbf{F}_{\mathrm{fins,I}}(t) + \mathbf{F}_{\mathrm{thr,I}}(t)\right] \tag{1}$$

where $\mathbf{F}_{\mathrm{aero,I}}(t)$ represents the aerodynamic force of the vehicle's body expressed in the ECI frame, $\mathbf{F}_{\mathrm{TVC,I}}(t)$, $\mathbf{F}_{\mathrm{fins,I}}(t)$ and $\mathbf{F}_{\mathrm{thr,I}}(t)$ represent control forces, $m(t)$ is the total mass of the vehicle and the gravity acceleration $\mathbf{g}_{\mathrm{I}}(t)$ is given by the Earth Gravitational Model [28] (EGM).

In addition, the rotational dynamics are described in the body-fixed frame by:

$$\dot{\omega}_{\mathrm{B}}(t) = J^{-1}(t) \left[\mathbf{M}_{\mathrm{aero,B}}(t) + \mathbf{M}_{\mathrm{TVC,B}}(t) + \mathbf{M}_{\mathrm{fins,B}}(t) + \mathbf{M}_{\mathrm{thr,B}}(t) - \omega_{\mathrm{B}}(t) \times J(t)\, \omega_{\mathrm{B}}(t) - \dot{J}(t)\, \omega_{\mathrm{B}}(t)\right] \tag{2}$$

Equivalently to the forces, $\mathbf{M}_{\mathrm{aero,B}}(t)$, $\mathbf{M}_{\mathrm{TVC,B}}(t)$, $\mathbf{M}_{\mathrm{fins,B}}(t)$ and $\mathbf{M}_{\mathrm{thr,B}}(t)$ represent aerodynamic and control moments written in the body axes, and $J(t)$ is the inertia tensor of the vehicle.

The orientation of the vehicle's body axes in the ECI frame is propagated through the kinematics equation:

$$\dot{\mathbf{q}}_{\mathrm{B}}^{\mathrm{I}}(t) = \frac{1}{2} \begin{bmatrix} q_4(t) & -q_3(t) & q_2(t) \\ q_3(t) & q_4(t) & -q_1(t) \\ -q_2(t) & q_1(t) & q_4(t) \\ -q_1(t) & -q_2(t) & -q_3(t) \end{bmatrix} \omega_{\mathrm{B}}(t) \tag{3}$$

where the quaternion $\mathbf{q}_{\mathrm{B}}^{\mathrm{I}}(t) = [q_1(t);\, q_2(t);\, q_3(t);\, q_4(t)]$ is assumed to have the scalar part, $q_4(t)$, as its last component. This vector is also essential for the computation of the forces and moments in Eq. (1) and (2). For a detailed description of these equations, the reader is referred to [24].

For the computation of aerodynamic forces and moments, a velocity reference frame is defined. It is fixed to the vehicles's centre of gravity (CG) and its x-axis is directed along the wind-relative velocity vector $\mathbf{v}_{\mathrm{air}}(t)$, so that the transformation from body-fixed to velocity reference frame can be represented by two aerodynamic angles, the angle of attack $\alpha(t)$ and sideslip $\beta(t)$, with respect to the pitch plane, also depicted in Fig. 1 (note that this plane changes slightly throughout the trajectory as a result of Earth's rotation). Aerodynamic characteristics depend on the vehicle's external shape and instantaneous dynamic pressure, which is given by:

$$Q(t) = \frac{1}{2} \rho(t) \left\| \mathbf{v}_{\mathrm{air}}(t) \right\|^2 \tag{4}$$

where $\mathbf{v}_{\mathrm{air}}(t)$ accounts for the vehicle's inertial velocity, Earth's rotation and wind gusts, and the air density $\rho(t)$ is

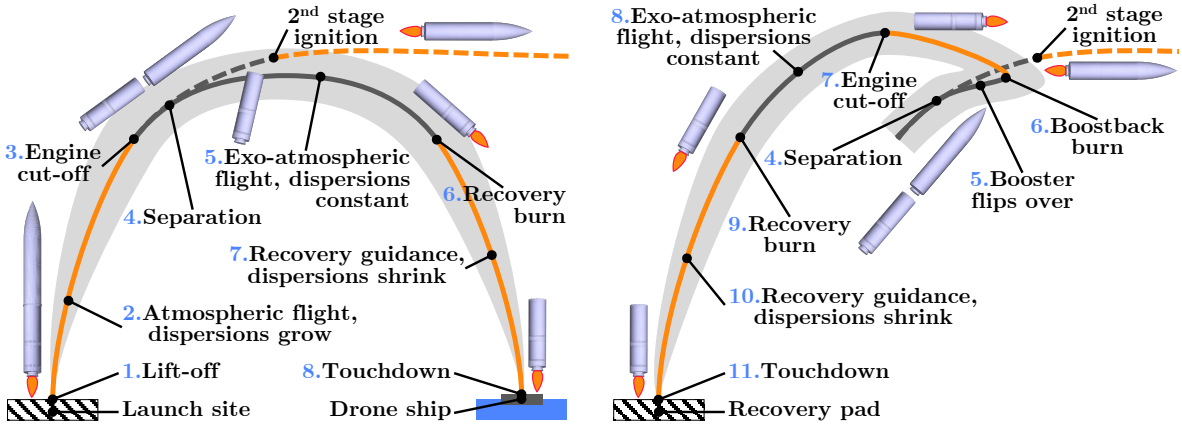
computed using the Committee on Extension to the Standard Atmosphere (COESA) model from [29]. Aerodynamic drag and lift coefficients, reference area and centre of pressure (CP) position are estimated from look-up tables as functions of the effective angle of attack $\alpha_{\text{eff}}(t) = \sqrt{\alpha^2(t) + \beta^2(t)}$ and Mach number.

The vehicle's structural properties are representative of Europe's lightweight VEGA launcher [30], but its first stage has been replaced by a fictional mixed-fuel booster. This booster uses a re-ignitable liquid-oxygen/kerosene (LOX/RP-1) rocket engine, with required propellant masses determined based on the reference mission under analysis. The mass-depletion dynamics due to propellant consumption is given by the rocket equation [27]:

$$\dot{m}(t) = -\frac{1}{I_{\text{sp}} g_0} T_{\text{ref}}(t) \tag{5}$$

where I_{sp} is the specific impulse of the engine, $g_0 \approx 9.81$ m/s^2 is the gravitational acceleration at the Earth's surface and $T_{\text{ref}}(t)$ is the magnitude of the thrust vector. In addition, CG position and inertia properties are updated throughout the flight based on the depletion of propellant and subsequent mass of oxygen and kerosene in the corresponding tanks.

Concerning the recovery of the vehicle, two distinct mission profiles are considered (summarised in Fig. 2): downrange landing (DRL), in which the RLV stage lands close to its un-propelled impact site, and return to launch site (RTLS), where the stage uses an additional firing to return to its launch site. Guidance calculations are made in a recovery pad (RP) reference frame, also indicated in Fig. 1, therefore guidance laws are defined in the same way for the DRL and RTLS scenarios. In both cases, a thrust vector is commanded by the guidance subsystem and then converted to reference pitch and yaw angles $\{\theta_{\text{ref}}(t), \psi_{\text{ref}}(t)\}$ and thrust magnitude $T_{\text{ref}}(t)$.



(a) DRL (b) RTLS

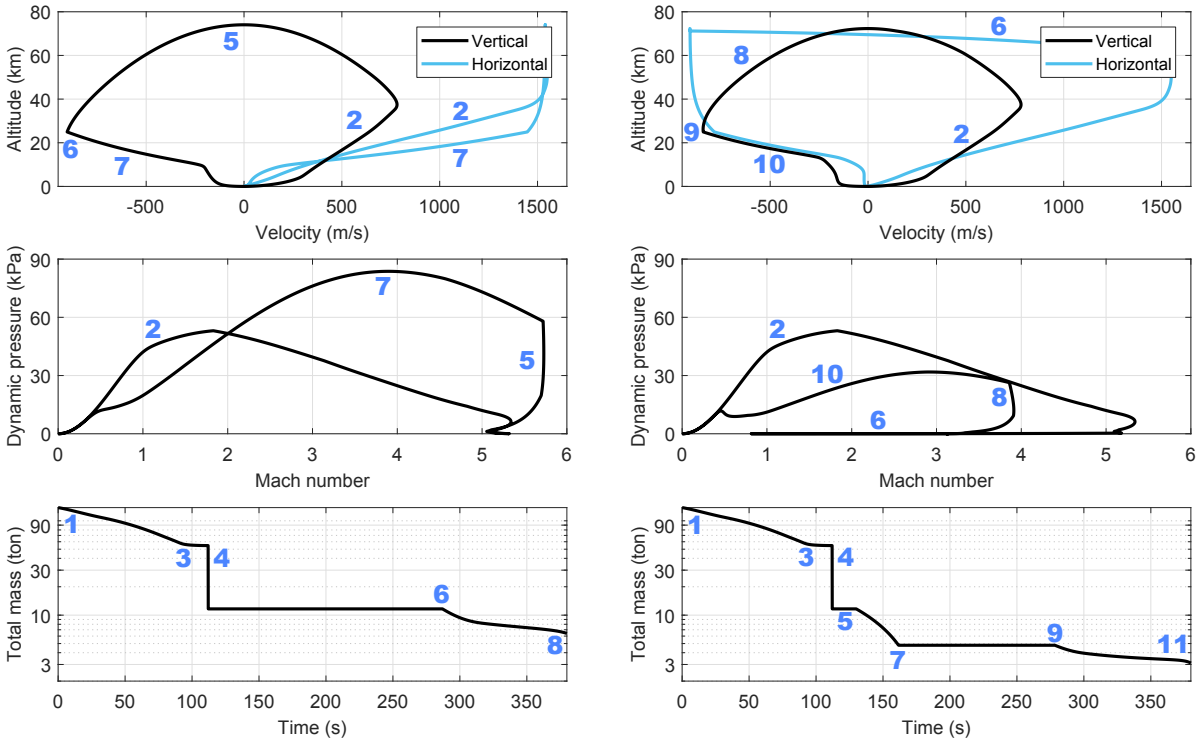
Fig. 2 Recovery mission profiles

Recovery guidance is triggered at a pre-specified altitude h_s, where the first stage re-ignites its engine to bring the booster from its current position and velocity to a soft touchdown at the recovery platform. Furthermore, in order to more efficiently meet aerothermal loads or propulsion system requirements, sometimes it is convenient to explicitly split

the recovery into two separate burns. In this case, the recovery consists of three sub-phases: (i) a re-entry burn aimed at decelerating the booster, (ii) a second engine cut-off and coast phase and (iii) a landing burn that ensures a precise touchdown. The duration of (i) and (ii) relative to the total recovery phase are defined by t_1 and t_2 respectively.

Thanks to its modular architecture, the benchmark allows for the modification of G&C algorithms and configuration/mission parameters, making it extremely versatile to study their interactions and effects at user-defined levels of fidelity (as required by the specific assessment objective).

With this versatility in mind, the present study employs a coarser aerodynamic coefficient set in order to reduce the computational load. Hence, the performance indicators provided in this article are slightly different from those introduced in [24], but the trends and conclusions established here have been verified to be independent of the aerodynamic fidelity level. An overview of the most relevant DRL and RTLS flight mechanics indicators using these coefficients is provided in Fig. 3, in which number labels relate the traces to the events of Fig. 2.



(a) DRL

(b) RTLS

Fig. 3 Comparison of the most relevant flight mechanics results

III. Improving Guidance via Flight Mechanics Trade-offs

A preliminary assessment of DRL and RTLS recovery strategies was carried out in [24] using a baseline guidance technique known as constrained terminal velocity (CTV). It was foreseen that the combined optimisation of distinct

performance indicators is an extremely challenging activity. For example, guidance choices that minimise propellant consumption are likely to subject the vehicle to higher aerothermal loads. Before tackling the improvement of performance through a more sophisticated guidance algorithm (Sec. IV), the purpose of this section is to analyse the impact of different guidance parameters and the compromises that can be achieved using the CTV technique. To do so, this technique is first recapped in Sec. III.A, then the performance-oriented analysis approach is introduced in Sec. III.B, and the subsequent results are provided and discussed in Sec. III.C.

A. Baseline guidance technique

CTV guidance has its roots in the missile interception problem [31], but it has been successfully applied also to asteroid intercept and landing [17–20]. The main strength of this technique lies in its simplicity, which makes it extremely easy to implement and provides a rough idea of recovery flight mechanics very quickly. It is based on the compensation of *zero-effort-miss* and *zero-effort-velocity* vectors, $\mathbf{ZEM}(t)$ and $\mathbf{ZEV}(t)$, which quantify the position and velocity error at the end-of-mission if no corrective manoeuvres are made after time t. Using these coordinates, the commanded thrust vector is given in the RP frame by:

$$\mathbf{T}_{\mathrm{CTV}}(t) = \hat{m}(t) \begin{bmatrix} k_r & k_v \end{bmatrix} \begin{bmatrix} \dfrac{\mathbf{ZEM}(t)}{(t_f - t)^2} \\ \\ \dfrac{\mathbf{ZEV}(t)}{t_f - t} \end{bmatrix} \tag{6}$$

This computation requires an estimate of the vehicle's mass $\hat{m}(t)$ and the specification of the end-of-mission (i.e. touchdown) time t_f. Optimal values of $\{6,\ -2\}$ for the two gains $\{k_r,\ k_v\}$ have been derived in [18, 19] by recasting the problem as a fuel-optimal trajectory generation problem with constrained boundary position and velocity and assuming a uniform and well-known gravity field, which is a valid approximation for the Earth.

The estimation of $\mathbf{ZEM}(t)$ and $\mathbf{ZEV}(t)$ involves propagating the equations of motion from t to t_f, which can become computationally challenging without some approximations. The ability of the guidance law to accurately enforce the boundary conditions (i.e. to minimise touchdown errors) naturally depends on the conservativeness of these approximations. The simplest possible guidance law can be obtained by neglecting: (i) gravity variations during descent, (ii) mass variations due to propellant consumption, (iii) aerodynamic forces, and (iv) non-inertial effects of the RP frame. Then, the resulting zero-effort errors correspond to:

$$\mathbf{ZEM}(t) = \mathbf{r}_f - \left[\hat{\mathbf{r}}(t) + (t_f - t)\,\hat{\mathbf{v}}(t) + \frac{1}{2}(t_f - t)^2\,\hat{\mathbf{g}}(t)\right] \tag{7}$$

$$\mathbf{ZEV}(t) = \mathbf{v}_f - \left[\hat{\mathbf{v}}(t) + (t_f - t)\,\hat{\mathbf{g}}(t)\right] \tag{8}$$

where $\mathbf{r}_f = \mathbf{v}_f = [0; 0; 0]$ for a soft landing at the recovery pad, and $\hat{\mathbf{r}}(t)$, $\hat{\mathbf{v}}(t)$ and $\hat{\mathbf{g}}(t)$ are position, velocity and gravity acceleration estimates in the RP frame. Although the vehicle's landing gear has to be able to withstand a (small) non-zero touchdown velocity, $\mathbf{v}_f$ is set to 0 m/s in order to be as conservative as possible. Care must also be taken in Eq. (6) to avoid a singularity when $t \rightarrow t_f$. The most effective way to do it is by switching-off the guidance commands immediately before the end-of-mission. The exact instant of time represents a trade-off between allowable touchdown error and maximum thrust authority.

Due to its simplicity, the computational time required by the CTV algorithm is extremely low, so guidance commands can be updated at the same frequency as the simulation, $f_{\text{gui}} = f_{\text{sim}}$, starting at a pre-specified altitude h_s. Initial values adopted for the verification of the algorithm are summarised in Table 1. The simulation frequency was set to 10 Hz since it was verified to provide enough accuracy for trajectory assessment while minimising simulation runtime.

Table 1 Initial CTV algorithm parameters

Parameter	Value
h_s (km)	25
t_f (s)	380
t_1 (%)	0
t_2 (%)	0
f_{sim} (Hz)	10
f_{gui} (Hz)	10

Because of the simplifications and algorithmic framework, the inherent capabilities of CTV guidance are rather limited. The most relevant limitations lie in its inability to explicitly enforce path constraints (i.e. only boundary states can be constrained) and account for mass-depletion dynamics. It is also noted that, although it is possible to use two burns for recovery (defined by t_1 and t_2), these burns are not explicitly accounted for with the CTV approach. Path constraints such as subsurface flight avoidance and bounded control capabilities are critical for RLV recovery, representing the main motivation behind the guidance algorithm developed in Sec. IV.

B. The trade-off map approach

The analysis approach adopted in this study was proposed in [32] as a methodology for space exploration guidance tuning, but also to provide an understanding of the performance trade-offs involved. It relies on systematic simulations of the nonlinear benchmark model over a guidance parameter grid to generate *trade-off maps* that enable a clear quantification of candidate choices.

Performance trade-off maps are generated by overlapping contour plots associated with key D&L metrics, which may represent either nominal or dispersed (e.g. standard deviation) values. For the present study, these indicators include:

- $m_{\text{rec}} = m(t_{\text{sep}}) - m(t_f)$, propellant mass required for recovery. This value is the most direct mission performance indicator since decreasing the required propellant enables an increase of payload mass and subsequent reduction of the launch cost per kilogram of payload;
- $Q_{\max} = \max Q(t)|_{t \in [t_{\text{sep}},\, t_f]}$, maximum dynamic pressure encountered during recovery, to which aerodynamic loads are proportional. A higher dynamic pressure therefore requires a more structurally robust vehicle's body to withstand the associated loads and typically leads to an increase of its dry mass;
- $v_f = ||\mathbf{v}(t_f)||$, touchdown velocity norm, which needs to be sustained by the vehicle's landing gear. Similar to the above, higher touchdown velocities will require reinforced mechanisms, which tend to increase the dry mass.

In the first two indicators, t_{sep} (=112 seconds) is the instant of time immediately after separation.

Another mission-critical indicator is the maximum thermal flux (proportional to $\|\mathbf{v}_{\text{air}}(t)\|^3$). This indicator was verified to follow similar trends to the maximum dynamic pressure (proportional to $\|\mathbf{v}_{\text{air}}(t)\|^2$, see Eq. (4)) and therefore it is not shown in the trade-off maps for the sake of conciseness.

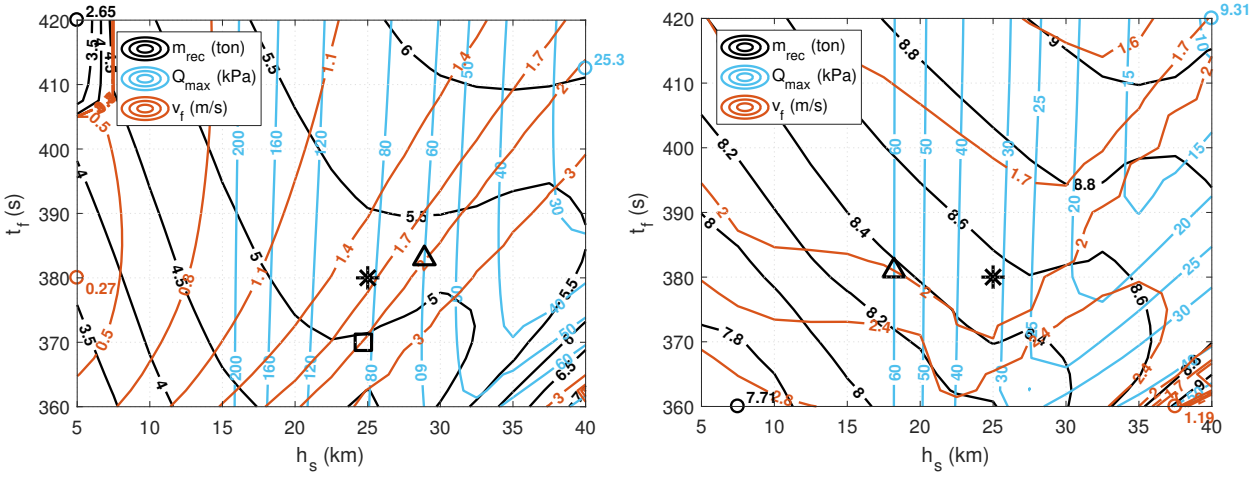
To conclude this section, it is important to recall that the two main parameters on which the CTV technique depends, h_s and t_f, are in fact intrinsically representative of the physics of the reusable problem. In other words, performance indicators will change depending on the guidance solution utilised, but their general trends remain comparable. Hence, CTV trade-off maps over the parameter space of h_s and t_f become extremely valuable since they can be generated very quickly and the understanding they provide regarding the performance impact of those parameters remains valid for other guidance algorithms.

C. Results and discussion

Trade-off maps for the DRL and RTLS missions are depicted in Fig. 4a and 4b, respectively. They both show contours of the previous indicators over the parameter space of start altitude $h_s \in [5,\, 40]$ km and final time $t_f \in [360,\, 420]$ seconds. In addition, the optimal (minimum) value of each indicator is highlighted using the symbol ◯ (with the colour corresponding to the appropriate indicator). Note that these optimal values lie on the axes of the figures.

Starting with DRL recovery, Fig. 4a, the trade-offs between propellant mass m_{rec} (black), dynamic pressure $Q_{\max}$ (cyan) and touchdown speed v_f (red) are clearly perceptible. For example, (i) the minimum m_{rec} (of 2.65 ton, located in the top-left corner) leads to a very high $Q_{\max}$ because it is associated with a very low start altitude, (ii) the minimum $Q_{\max}$ (of 25.3 kPa, on the top-right side) is related to a high m_{rec}, as physically expected, and (iii) the minimum v_f (of 0.27 m/s, located towards the mid-bottom of the left axis) requires more demanding guidance commands – mostly via a reduced final time (recall Eq. (6)) and therefore increased m_{rec} and $Q_{\max}$.

In other words, a single guidance choice cannot simultaneously attain all three optima, but is the result of a performance compromise. Suitable compromises and guidance choices can however be easily identified and quantified using the trade-off map.



(a) DRL (b) RTLS

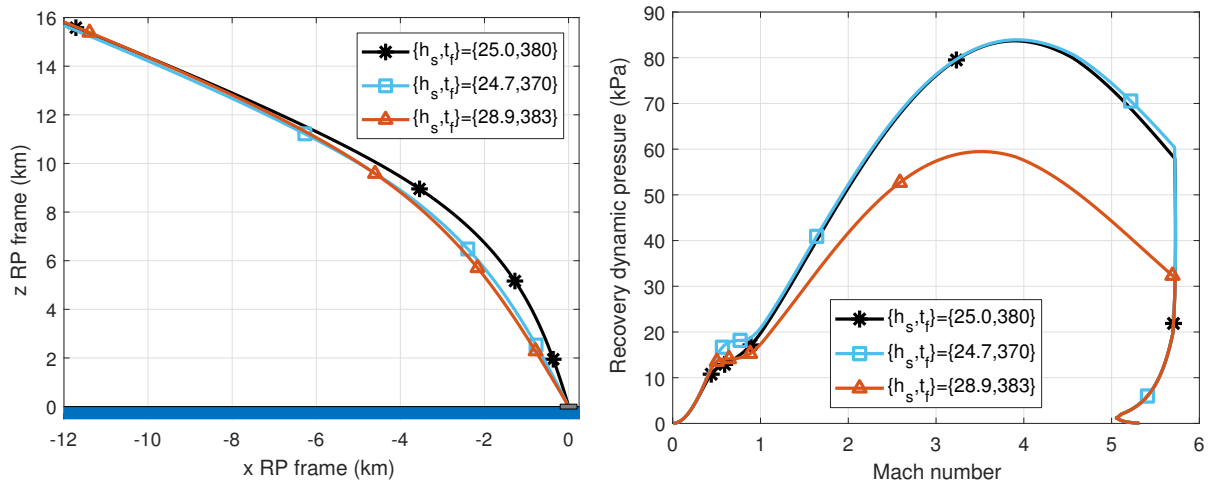
Fig. 4 CTV Trade-off maps for h_s and t_f

As an illustration, consider the initial choice taken from Table 1 of $\{h_s,\ t_f\} = \{25.0,\ 380\}$ (indicated by the symbol $*$ in Fig. 4a). This choice leads to $m_{\text{rec}} = 5.26$ ton, $Q_{\max} = 83.7$ kPa and $v_f = 1.62$ m/s. Now assume that the vehicle's landing gear has a stronger structural resistance and is able to withstand speeds 25% higher, up to 2 m/s (meaning that it does not need to slow down as much throughout the descent). In this case, keeping the same value of $Q_{\max}$, the required propellant can be reduced to 4.95 ton by choosing $\{h_s,\ t_f\} = \{24.7,\ 370\}$ (indicated by the symbol $\square$). This choice translates to initiating the manoeuvre slightly later and finishing 10 seconds earlier. If, in addition to that, the vehicle is only able to sustain a dynamic pressure of 60 kPa (closer to the 53 kPa encountered during launch), the optimal guidance choice becomes $\{h_s,\ t_f\} = \{28.9,\ 383\}$ (marked by $\triangle$) and the required propellant increases to 5.30 ton.

It is interesting to note that, although the latter guidance choice ($\triangle$) and the initial one ($*$) have comparable propellant requirements, they are considerably different in terms of trajectory and dynamic pressure, as evidenced by Fig. 5a and 5b, respectively. This consideration proves that it is not enough to rely on a single indicator for a fair comparison of D&L guidance approaches.

The trade-off map for RTLS recovery is then provided in Fig. 4b. Performance trends are roughly similar to the DRL case, although values of m_{rec} are now globally higher and values of $Q_{\max}$ are lower. This effect is caused by the additional firing needed to bring the stage back, which additionally reduces the magnitude of its horizontal velocity and thus dynamic pressure. From [24], it is also known that only about 8.9 ton of propellant is available after launch and therefore not all the parameter space shown in the trade-off map is feasible.

Conflicting guidance choices that optimise each single indicator for the RTLS case can again be identified: (i) minimum m_{rec} (of 7.71 ton, bottom-left corner) leads to a high $Q_{\max}$, (ii) minimum $Q_{\max}$ (of 9.31 kPa, top-right corner) is related to a high m_{rec}, and (iii) minimum v_f (of 1.19 m/s, bottom-right corner) lies in one of the infeasible



(a) Trajectory in the RP frame

(b) Dynamic pressure vs. Mach number

Fig. 5 DRL results with different guidance choices (markers show increments of 20 seconds)

areas mentioned above. The same trade-off exercise carried out for DRL recovery is repeated next.

The initial choice (*) from Table 1 of $\{h_s,\ t_f\} = \{25.0,\ 380\}$ leads to $m_{\text{rec}} = 8.55$ ton, $Q_{\max} = 31.8$ kPa and $v_f = 1.85$ m/s. Now consider that the vehicle's constraints are exactly the same as for the $\triangle$ point before, i.e. 60 kPa dynamic pressure and 2 m/s touchdown speed. In this case, revising the guidance choice to $\{h_s,\ t_f\} = \{18.2,\ 381\}$ allows an alleviation of the propellant requirement to 8.37 ton (also marked in Fig. 4b by $\triangle$).

IV. Improving Guidance via Convex Optimisation

Convex optimisation guidance is based on solving a fuel-optimal trajectory generation problem with state and control constraints. This problem is typically nonlinear and challenging to solve and, until the past decade, its application was only feasible offline (where open-loop trajectories are designed on the ground with powerful computers). However, mathematical and computational developments in recent years have enabled representative solutions to be determined online using onboard computers and applied in a closed-loop fashion [8]. This approach, where an optimal control input is computed based on the predicted trajectory, enables tackling the D&L problem in a model predictive control (MPC) setting [33–35].

The most relevant mathematical developments (lossless and successive convexification [9–12]) are briefly introduced in Sec. IV.A. Based on these developments (and taking into account the previous trade-off map assessment), an algorithm that is suitable for the extended flight envelope encountered by RLVs is proposed in Sec. IV.B and detailed in Sec. IV.C.

A. Lossless and successive convexification

Lossless convexification (refer to [9, 10] for details) is a procedure that can be used to relax non-convex constraints into a convex form, and then prove the equivalence of the resulting optimal control problem. The constrained fuel-optimal

trajectory generation problem, once discretised, becomes a second-order cone program (SOCP), for which powerful interior-point solvers exist [36, 37].

In addition, a technique known as successive convexification [11, 12] can be applied to approximate any remaining nonlinearities such as aerodynamic effects. This technique constitutes an iterative process in which the nonlinearities are repeatedly linearised using information from the previous solution.

These techniques represent the foundation of the powered D&L algorithm developed in this study. The proposed algorithm is based specifically on the work of *Szmuk et al.* [25] and *Jerez et al.* [26], with a few fundamental and critical differences. The latter reference [26] is mostly focused on maximising computational efficiency. Although ideal for real-time execution, the algorithm in [26] relies on an extensive simplification of the equations of motion including disregarding aerodynamic forces, which play a decisive role in the vehicle's recovery trajectory. On the other hand, references [25] take a completely different approach to [26] and focus on maximising the optimality of the solution by employing successive convexifications to account for aerodynamic effects (as well as engine back-pressure losses). It is also noted that the two approaches have only been verified for low-altitude and low-velocity flight and that the assumption of constant air density made in [25] is not physically representative of RLVs.

With these considerations in mind, the objective of the proposed algorithm is to attain a middle ground between the efficiency and optimality of the two aforementioned approaches that is suitable for the extended flight envelope encountered by RLVs. This algorithm is termed *DESCENDO* (Descending over Extended Envelopes using Successive Convexification-based Optimisation).

One additional difference with respect to *Szmuk et al.* [25] is that the touchdown time t_f is not an optimisation variable and needs to be specified. This choice is made in order to provide a common comparative framework with the baseline approach of Sec. III.A. Moreover, translational and rotational motions can be addressed in combination using successive convexification [38] or a dual-quaternion representation [39]. However, this study relies on a perfect attitude control assumption, which means that attitude angles are exactly what they are commanded to be. This is a common practice for developing and assessing guidance schemes since attitude can be changed quickly compared to the trajectory. The impact of attitude control and its minimisation using robust load prediction and relief algorithms are currently being investigated under the scope of the same activity.

B. The *DESCENDO* algorithm strategy

The *DESCENDO* guidance algorithm is schematised in Fig. 6 and further detailed in Sec. IV.C. It has been implemented in MATLAB using the CVX library [40] to formulate the convex problems, and the ECOS routine [41] to solve them. The proposed algorithm consists of two SOCP stages, see Fig. 6: SOCP 1, which allows to find a discrete trajectory and acceleration profile that do not account for aerodynamic effects, and SOCP 2 where successive convexifications are iteratively applied to define a convex approximation of those effects.

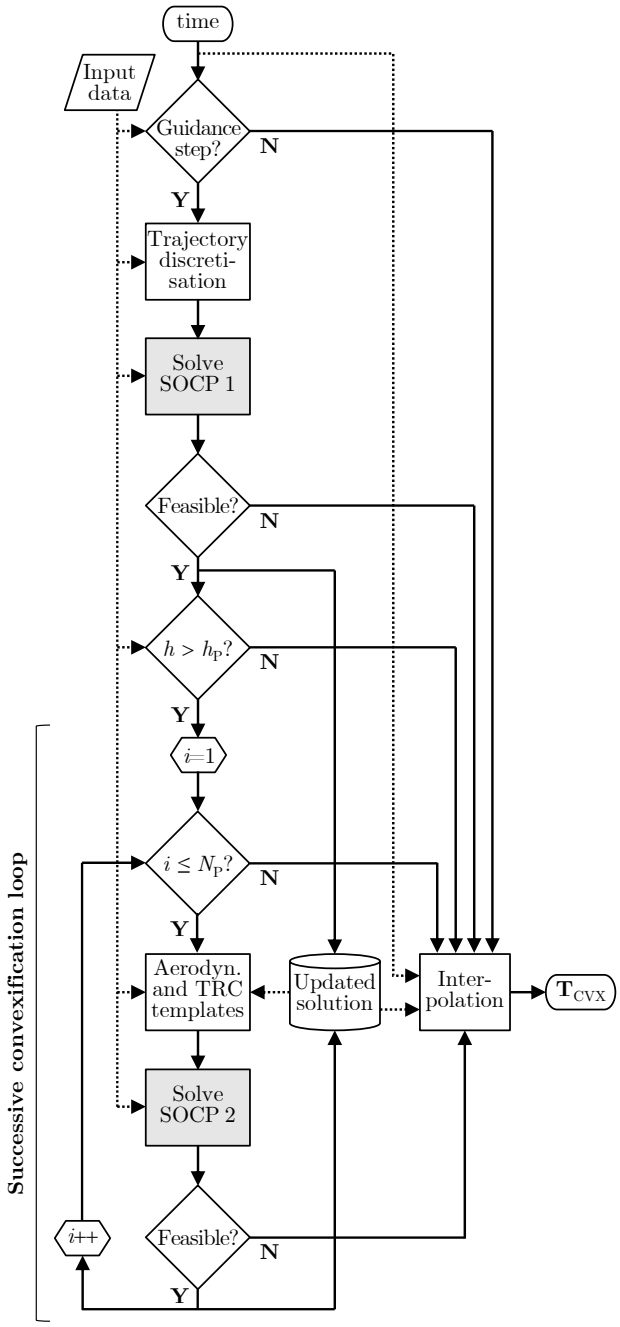


Fig. 6 *DESCENDO* guidance algorithm

At each simulation instance (determined by the simulation rate f_{sim}), the commanded thrust vector in the RP frame $\mathbf{T}_{\text{CVX}}(t)$ is computed from the most recent guidance solution (via linear interpolation). That solution is stored as an online look-up table and updated only when an SOCP step is executed and a feasible solution is found.

SOCP 1 is triggered at every *guidance step*, which is determined by the guidance update frequency f_{gui} and by a pre-specified altitude h_s. To enable its formulation, trajectory and optimisation variables are first discretised into N

uniformly-spaced points, ranging from the current instant of time t to touchdown time t_f. The discretisation interval between two consecutive points corresponds to:

$$T_S = \frac{t_f - t}{N - 1} \tag{9}$$

and, since $T_S \to 0$ as $t \to t_f$, the accuracy of the discretisation becomes more refined towards the end. The execution of SOCP 2 is determined by two additional variables, h_P and N_P (not to be confused with h_s and N), which will be further detailed in Sec. IV.C.

The accuracy of *DESCENDO* increases with the guidance update frequency f_{gui} and with the number of points N, at the expense of a higher computational load (for this reason, f_{gui} is typically smaller than the simulation rate f_{sim}). It is important to note that a larger f_{gui} does not increase the accuracy of the SOCPs themselves, but executing them more frequently minimises errors introduced by the trajectory discretisation and aerodynamic forces.

N can be adjusted over time, in order to have a larger number of points in earlier (longer) trajectories. In practice, varying N is more demanding as it involves the implementation and validation of different guidance modes. Since this preliminary study is focused on the behaviour of the algorithm itself, the extra effort will be avoided by keeping N constant.

C. The *DESCENDO* algorithm

The fuel-optimal trajectory generation problem contains two sources of non-convexity that can be tackled with the principle of lossless convexification: (i) the propellant-depletion dynamics, which has a logarithmic dependence on time as evidenced by Eq. (5), and (ii) thrust magnitude and pointing constraints, due to the norm operator. As introduced in [9], the first one is directly convexified using the following change of coordinates:

$$z(t) = \ln \hat{m}(t), \qquad \dot{z}(t) = \frac{\dot{\hat{m}}(t)}{\hat{m}(t)} \tag{10}$$

while the latter requires the introduction of an additional optimisation constraint defined using two new variables:

$$\mathbf{w}(t) = \frac{\mathbf{T}_{CVX}(t)}{\hat{m}(t)}, \qquad \sigma(t) = \frac{\|\mathbf{T}_{CVX}(t)\|}{\hat{m}(t)} \tag{11}$$

Using this change of variables and the setup of Fig. 6, the objective of the first SOCP is to find a discrete thrust acceleration profile $\mathbf{w}[k]$ ($k \in [1, \cdots, N]$) that minimises the vehicle's fuel consumption, which is equivalent to maximising its final mass or $z[N]$. This specific problem is well established based on [9, 25, 26] and formally defined as:

$$\min_{\mathbf{w},\sigma} \sum_{k=1}^{N} \sigma[k] \;=\; \max_{\mathbf{w},\sigma} z[N] \tag{12}$$

Its main constraints are described in the following paragraphs.

Similar to CTV guidance, the optimisation problem is subject to initial and final boundary states at $k = 1$ and $k = N$, respectively. The former set up the current mass, position, velocity and thrust acceleration, i.e. $z[1] = \ln \hat{m}(t)$, $\mathbf{r}[1] = \hat{\mathbf{r}}(t)$, $\mathbf{v}[1] = \hat{\mathbf{v}}(t)$ and $\mathbf{w}[1] = \hat{\mathbf{w}}(t)$, while the latter specify the conditions at touchdown. In this case, (i) the final position must coincide with the landing site, $\mathbf{r}[N] = \mathbf{r}_f$, (ii) the final horizontal velocity with the desired value, $\mathbf{v}_{x,y}[N] = \mathbf{v}_{f_{x,y}}$, (iii) the vertical velocity must be smaller than a certain safety margin, $\mathbf{v}_z[N] \leq \mathbf{v}_{f_z}$, and (iv) the final thrust acceleration vector is required to have a positive vertical component only, $\mathbf{w}_{x,y}[N] = 0_{2\times1}$ and $\mathbf{w}_z[N] \geq 0$, so that the vehicle lands with an upright orientation.

The optimisation problem is also subject to the dynamics equations that dictate the time-evolution of the aforementioned states. These equations are discretised using time-interval T_S (Eq. (9)) and the knowledge that acceleration is linearly interpolated between two consecutive points. The first set of equations represents the translational motion of Eq. (1). As mentioned before, recovery calculations are made in an RP reference frame where non-inertial effects are neglected, hence the discrete equations correspond to:

$$\begin{aligned} \mathbf{r}[k+1] &= \mathbf{r}[k] + T_\mathrm{S}\,\mathbf{v}[k] + \frac{T_\mathrm{S}^2}{3}\left(\mathbf{a}[k] + \frac{\mathbf{a}[k+1]}{2}\right) \\ \mathbf{v}[k+1] &= \mathbf{v}[k] + \frac{T_\mathrm{S}}{2}\left(\mathbf{a}[k] + \mathbf{a}[k+1]\right) \end{aligned} \tag{13}$$

where $\mathbf{r}[k]$, $\mathbf{v}[k]$ and $\mathbf{a}[k]$ represent position, velocity and acceleration in the RP frame. The second equation describes the mass-depletion dynamics of Eq. (5) which, following the change of variables of Eq. (10) and (11) becomes:

$$z[k+1] = z[k] - \frac{1}{I_\mathrm{sp} g_0}\frac{T_\mathrm{S}}{2}\left(\sigma[k] + \sigma[k+1]\right) \tag{14}$$

The surrogate variables $\mathbf{a}[k]$ and $\sigma[k]$ are then defined. The former variable gathers the vehicle's acceleration contributions:

$$\mathbf{a}[k] = \mathbf{w}[k] + \hat{\mathbf{g}}(t) \tag{15}$$

where $\mathbf{w}[k]$ is given by Eq. (11) and the gravity acceleration $\hat{\mathbf{g}}(t)$ is assumed constant from time t onwards. At this point, aerodynamic effects are not yet included. In addition, the inequality:

$$\|\mathbf{w}[k]\| \leq \sigma[k] \tag{16}$$

is introduced as part of lossless convexification procedure [9, 10], with $\sigma[k] \to \|\mathbf{w}[k]\|$ when $z[N]$ is maximised.

In addition, control constraints are employed to bound the direction and magnitude of the thrust force. The direction constraint indirectly limits the angle between the vehicle's longitudinal axis and the vertical direction to θ_max via:

$$\mathbf{w}_z[k] \geq \frac{\|\mathbf{w}_{x,y}[k]\|}{\tan\theta_{\max}} \tag{17}$$

Lower and upper thrust magnitude limits are given by $\{T_{\min}, T_{\max}\}$. To preserve convexity, constant mass $\hat{m}(t)$ is assumed from t onwards, which is not restrictive since the actual limit values can be re-adjusted, and the constraint becomes [9]:

$$\frac{T_{\min}}{\hat{m}(t)} \leq \sigma[k] \leq \frac{T_{\max}}{\hat{m}(t)} \tag{18}$$

Constraints (17) and (18) are only enforced during the burn periods defined by a pre-specified interval $\mathcal{T}_{\mathrm{P}}$, which depend on the relative duration of re-entry and landing burn, t_1 and t_2. Outside these periods, the applied thrust acceleration is set to zero.

The maximum thrust magnitude rate is bounded to $\dot{T}_{\max}$ using a forward discretisation scheme for the differentiation of $\sigma[k]$ and a constraint equivalent to Eq. (18):

$$\sigma[k] - T_{\mathrm{S}}\frac{\dot{T}_{\max}}{\hat{m}(t)} \leq \sigma[k+1] \leq \sigma[k] + T_{\mathrm{S}}\frac{\dot{T}_{\max}}{\hat{m}(t)} \tag{19}$$

This constraint was proven critical to the reduction of control chattering without having a noticeable impact on the tightness of Eq. (16).

The optimisation problem subject to the constraints introduced above is then formulated in SOCP 1.

The main limitation of this formulation lies in its inability to account for aerodynamic forces. Without any knowledge of the deceleration caused by these forces, the algorithm estimates that more propellant is needed to slow the vehicle down than it is in reality. But most importantly, it was observed that, because of this discrepancy, earlier SOCP 1 solutions could only be found if subsurface flight avoidance (or any other flight path constraint) is not enforced. Moreover, the algorithm was able to recover more effectively from an earlier solution where all boundary conditions are met (even if with subsurface flight) than from one without subsurface flight. This recovery will be highlighted in Sec. V.

To overcome the aforementioned limitations, the *DESCENDO* algorithm then introduces the successive convexification procedure of [11, 12]. This procedure involves solving a second (iteratively more refined) SOCP in which the solution of the previous problem is employed to define a *convex approximation* of the aerodynamic effects. This cycle is executed N_{P} times *per guidance step*, and the solution of SOCP 1 (if feasible) is used for the first iteration. This approach results in $1 + N_{\mathrm{P}}$ SOCPs (i.e. 1 SOCP 1 and N_{P} SOCPs 2) being solved at each guidance step. If SOCP 1 is not feasible, the guidance solution is not updated and the interpolation uses the solution obtained in the previous guidance step.

SOCP 1

$$\max_{\mathbf{w},\sigma} \; z[N], \quad \text{subject to:}$$

Boundary conditions

$$z[1] = \ln \hat{m}(t), \quad \mathbf{r}[1] = \hat{\mathbf{r}}(t), \quad \mathbf{v}[1] = \hat{\mathbf{v}}(t), \quad \mathbf{w}[1] = \hat{\mathbf{w}}(t)$$

$$\mathbf{r}[N] = \mathbf{r}_f, \quad \mathbf{v}_{x,y}[N] = \mathbf{v}_{f_{x,y}}, \quad \mathbf{v}_z[N] \leq \mathbf{v}_{f_z}, \quad \mathbf{w}_{x,y}[N] = 0_{2\times 1}, \quad \mathbf{w}_z[N] \geq 0$$

Dynamics equations, $\forall k \in [1, \cdots, N-1]$

$$\mathbf{r}[k+1] = \mathbf{r}[k] + T_\mathrm{S}\,\mathbf{v}[k] + \frac{T_\mathrm{S}^2}{3}\left(\mathbf{a}[k] + \frac{\mathbf{a}[k+1]}{2}\right)$$

$$\mathbf{v}[k+1] = \mathbf{v}[k] + \frac{T_\mathrm{S}}{2}\left(\mathbf{a}[k] + \mathbf{a}[k+1]\right)$$

$$z[k+1] = z[k] - \frac{1}{I_\mathrm{sp} g_0}\frac{T_\mathrm{S}}{2}\left(\sigma[k] + \sigma[k+1]\right)$$

Surrogate variables, $\forall k \in [1, \cdots, N]$

$$\mathbf{a}[k] = \mathbf{w}[k] + \hat{\mathbf{g}}(t)$$

$$\|\mathbf{w}[k]\| \leq \sigma[k]$$

Control constraints, $\forall k \in [1, \cdots, N-1]$

$$\begin{cases} \mathbf{w}_z[k] \geq \dfrac{\|\mathbf{w}_{x,y}[k]\|}{\tan\theta_\mathrm{max}}, \quad \dfrac{T_\mathrm{min}}{\hat{m}(t)} \leq \sigma[k] \leq \dfrac{T_\mathrm{max}}{\hat{m}(t)}, & \text{if } T_\mathrm{S}(k-1) \in \mathcal{T}_\mathrm{P} \\ \mathbf{w}[k] = 0_{3\times 1}, & \text{otherwise} \end{cases}$$

Control rate constraints, $\forall k \in [1, \cdots, N-1]$

$$\sigma[k] - T_\mathrm{S}\frac{\dot{T}_\mathrm{max}}{\hat{m}(t)} \leq \sigma[k+1] \leq \sigma[k] + T_\mathrm{S}\frac{\dot{T}_\mathrm{max}}{\hat{m}(t)}$$

Aerodynamic effects in the refined SOCP 2 are approximated by augmenting the surrogate acceleration vector with a velocity-dependent term defined as:

$$\mathbf{a}[k] = \mathbf{w}[k] + \hat{\mathbf{g}}(t) - d_i^*[k]\mathbf{v}[k] \tag{20}$$

where $d_i^*[k]$ is a template computed before the SOCP, and therefore its complexity does not affect the efficiency of the problem's solution. Since the most significant aerodynamic effect during recovery is the deceleration due to drag, which is parallel to the velocity vector, $d_i^*[k]$ is defined as:

$$d_i^*[k] = \frac{1}{2}\rho_i^*[k]\, S_\mathrm{ref} C_{Di}^*[k]\, \frac{\|\mathbf{v}_i^*[k]\|}{\exp z_i^*[k]} \tag{21}$$

In this equation, $\mathbf{v}_i^*[k]$ and $z_i^*[k]$ are given directly by the solution of the previous SOCP i^{th} iteration, and $\rho_i^*[k]$ and $C_{Di}^*[k]$ can be estimated as functions of $\mathbf{v}_i^*[k]$ and $\mathbf{r}_i^*[k]$. For this scenario however, defining $\rho_i^*[k]$ as a linearly-spaced vector from $\hat{\rho}(t)$ to ρ_0 ($\rho_0 \approx 1.23$ kg/m^3 is the density at sea level) and $C_{Di}^*[k] = C_D|_{\alpha=\pi}$ (i.e. assuming that the vehicle's downward velocity vector is aligned with its longitudinal axis, recall Fig. 1) as a constant was verified to yield acceptable results.

In addition to the aerodynamic template, the iterative process includes a condition that facilitates the algorithm's convergence by bounding the deviation between guidance solutions found in two consecutive iterations. This condition is known as *trust region constraint* (TRC), see also Fig. 6, and defined as:

$$\|\mathbf{w}[k] - \mathbf{w}_i^*[k]\| \leq \eta_{\mathbf{w}}[k] \tag{22}$$

where $\mathbf{w}_i^*[k]$ is the thrust acceleration template determined by the previous SOCP i^{th} iteration. The TRC is enforced by minimising $\eta_{\mathbf{w}}[k]$, hence the SOCP objective function needs to be augmented with the point-wise sum of this vector, weighted by $w_{\eta_{\mathbf{w}}}$. A smaller value of $w_{\eta_{\mathbf{w}}}$ will be reflected in a larger variation between solutions and vice-versa.

The formulation of the refined optimisation problem featuring the acceleration of Eq. (20), the TRC of Eq. (22) and the augmented objective function is provided in SOCP 2.

Comparing this problem with SOCP 1, two increasingly stringent specifications are made: (i) the inequality relaxation of the final vertical velocity $\mathbf{v}_z[N]$ is dropped, and (ii) the following flight path constraint is introduced:

$$\mathbf{r}_z[k] \geq \frac{\hat{\mathbf{r}}_z(t)}{\|\hat{\mathbf{r}}_{x,y}(t)\|}\|\mathbf{r}_{x,y}[k]\| \tag{23}$$

In addition to subsurface flight avoidance, this constraint ensures that the recovery trajectory remains in the interior of a shrinking cone with vertex at the landing point and with the vehicle's current position $\hat{\mathbf{r}}(t)$ on its surface.

It is also important to note that, as evidenced in Fig. 6, SOCP 2 is only solved while the vehicle is higher than a pre-specified altitude h_{P}. The reason for this choice is related to the fact that velocity is significantly smaller at low altitudes and thus aerodynamic forces become less intense. Therefore, disregarding their impact here introduces less error while reducing the overall computational time.

SOCP 2

$$\max_{\mathbf{w},\sigma} \; z[N] - w_{\eta_\mathbf{w}} \sum_{k=1}^{N} \eta_\mathbf{w}[k], \quad \text{subject to:}$$

Boundary conditions

$$z[1] = \ln \hat{m}(t), \quad \mathbf{r}[1] = \hat{\mathbf{r}}(t), \quad \mathbf{v}[1] = \hat{\mathbf{v}}(t), \quad \mathbf{w}[1] = \hat{\mathbf{w}}(t)$$

$$\mathbf{r}[N] = \mathbf{r}_f, \quad \mathbf{v}[N] = \mathbf{v}_f, \quad \mathbf{w}_{x,y}[N] = 0_{\,2\times1}, \quad \mathbf{w}_z[N] \geq 0$$

Dynamics equations, $\quad \forall k \in [1, \cdots, N-1]$

$$\mathbf{r}[k+1] = \mathbf{r}[k] + T_\mathrm{S}\,\mathbf{v}[k] + \frac{T_\mathrm{S}^2}{3}\left(\mathbf{a}[k] + \frac{\mathbf{a}[k+1]}{2}\right)$$

$$\mathbf{v}[k+1] = \mathbf{v}[k] + \frac{T_\mathrm{S}}{2}\left(\mathbf{a}[k] + \mathbf{a}[k+1]\right)$$

$$z[k+1] = z[k] - \frac{1}{I_\mathrm{sp} g_0}\frac{T_\mathrm{S}}{2}\left(\sigma[k] + \sigma[k+1]\right)$$

Surrogate variables, $\quad \forall k \in [1, \cdots, N]$

$$\mathbf{a}[k] = \mathbf{w}[k] + \hat{\mathbf{g}}(t) - d_i^*[k]\mathbf{v}[k]$$

$$\|\mathbf{w}[k]\| \leq \sigma[k]$$

Trust region constraints, $\quad \forall k \in [1, \cdots, N]$

$$\|\mathbf{w}[k] - \mathbf{w}_i^*[k]\| \leq \eta_\mathbf{w}[k]$$

Flight path constraints, $\quad \forall k \in [1, \cdots, N-1]$

$$\mathbf{r}_z[k] \geq \frac{\hat{\mathbf{r}}_z(t)}{\|\hat{\mathbf{r}}_{x,y}(t)\|}\|\mathbf{r}_{x,y}[k]\|$$

Control constraints, $\quad \forall k \in [1, \cdots, N-1]$

$$\begin{cases} \mathbf{w}_z[k] \geq \dfrac{\|\mathbf{w}_{x,y}[k]\|}{\tan\theta_\mathrm{max}}, \quad \dfrac{T_\mathrm{min}}{\hat{m}(t)} \leq \sigma[k] \leq \dfrac{T_\mathrm{max}}{\hat{m}(t)}, & \text{if } \; T_\mathrm{S}(k-1) \in \mathcal{T}_\mathrm{P} \\ \mathbf{w}[k] = 0_{\,3\times1}, & \text{otherwise} \end{cases}$$

Control rate constraints, $\quad \forall k \in [1, \cdots, N-1]$

$$\sigma[k] - T_\mathrm{S}\frac{\dot{T}_\mathrm{max}}{\hat{m}(t)} \leq \sigma[k+1] \leq \sigma[k] + T_\mathrm{S}\frac{\dot{T}_\mathrm{max}}{\hat{m}(t)}$$

V. Results and discussion of the *DESCENDO* algorithm

The initial parameters adopted for the verification of the *DESCENDO* algorithm are listed in Table 2. The start altitude (h_s) and final time (t_f) are maintained for consistency the same as for the baseline guidance in Sec. III.A. The thrust control constraints are set to $T_\mathrm{max} = 600$ kN and $\dot{T}_\mathrm{max} = 1$ kN/s, while T_min is set to zero without loss of generality.

Finally, values of h_{P}, $\theta_{\max}$ and $w_{\eta_{\mathrm{w}}}$ are stipulated as a result of a pattern search aimed at minimising $Q_{\max}$ while keeping computational times as low as possible. It is remarked that testing optimisation-based algorithms in closed-loop is particularly challenging due to the drastic influence of the choice of f_{gui} and N. These parameters have been kept constant throughout the present study, but a thorough assessment of their impact would be crucial prior to any real-world application.

Table 2 Initial *DESCENDO* algorithm parameters

Parameter	Value
h_s (km)	25
t_f (s)	380
t_1 (%)	0
t_2 (%)	0
f_{sim} (Hz)	10
f_{gui} (Hz)	1
N	20

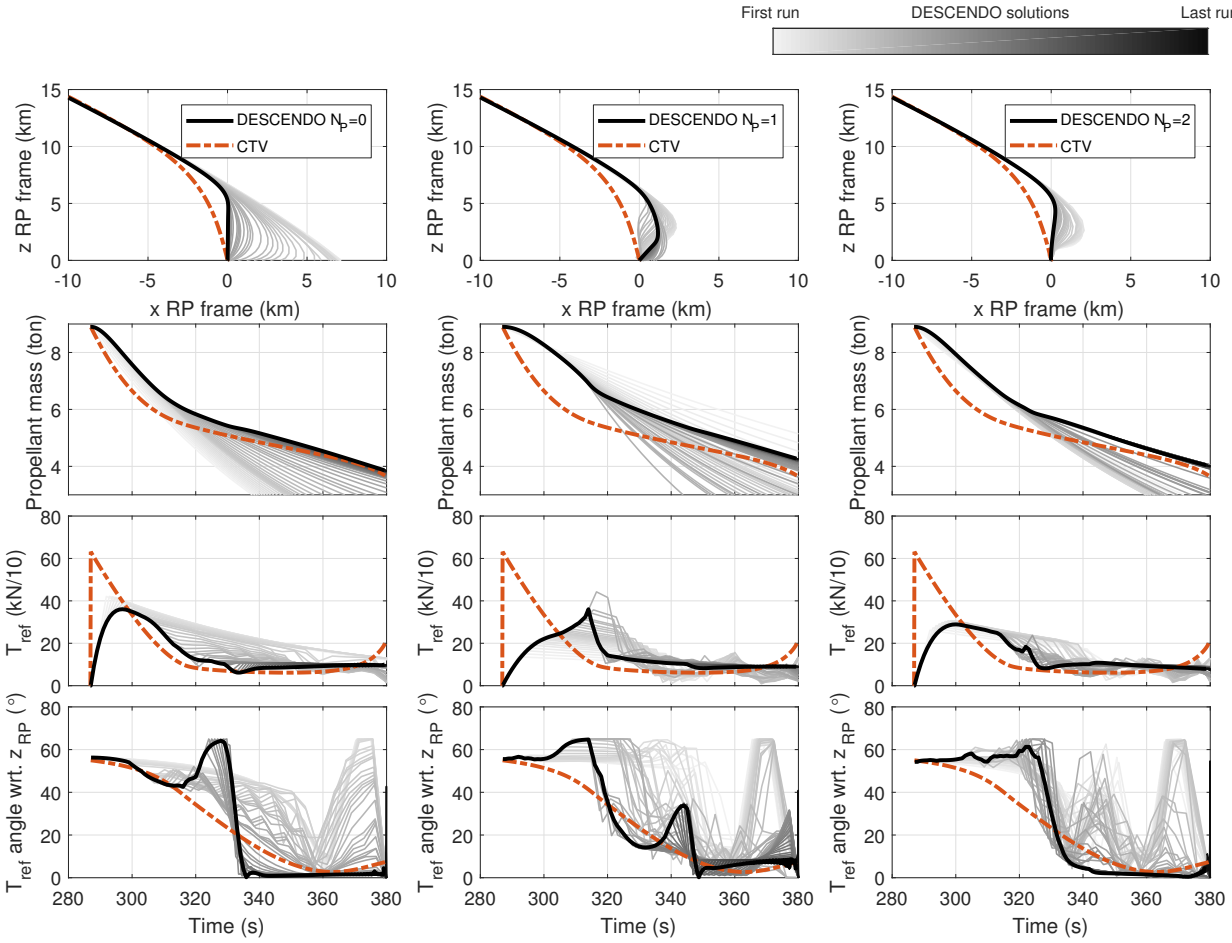
Parameter	Value
N_{P}	$\{0, 1, 2\}$
h_{P} (km)	2
$\theta_{\max}$ (deg)	65 (DRL), 70 (RTLS)
$T_{\min}$ (kN)	0
$T_{\max}$ (kN)	600
$\dot{T}_{\max}$ (kN/s)	1
$w_{\eta_{\mathrm{w}}}$	0.0001

Figure 7 gathers a set of closed-loop DRL simulations using *DESCENDO* and its three columns (Fig. 7a, 7b and 7c) correspond to solutions with $N_{\mathrm{P}} = \{0, 1, 2\}$. The main purpose of this analysis is to demonstrate the positive impact of having merely 1 or 2 successive convexification loops, rather than showing how accurate the algorithm can be for a larger number (which, in practice, is only limited by the available computational power).

The rows of Fig. 7 represent trajectory in the RP frame (similar to Fig. 5a), remaining propellant mass, thrust magnitude, and angle with respect to the vertical direction. Furthermore, each plot shows the CTV baseline in red (which is the same for all the values of N_{P}), the *DESCENDO* result in black and its intermediate solutions (i.e. those used for interpolation, recall Fig. 6) in a grey-scale. It is important to note that lighter colours are associated with solutions earlier in the descent. All these solutions start from their current states in the black line and become more accurate towards touchdown as $T_{\mathrm{S}} \to 0$ (with the same number of points N spanning over shorter distances).

Figure 7a shows that a successful recovery is already achieved with $N_{\mathrm{P}} = 0$. This outcome can be interpreted as the result of an algorithm in which aerodynamic effects are not considered, such as [26]. However, without any knowledge of the deceleration caused by aerodynamic forces, the algorithm provides inadequate guidance solutions earlier in the descent, which is demonstrated by the large dispersion of intermediate trajectory and mass estimates. Although these estimates are not satisfactory since they assume subsurface flight (as explained before) and often require more propellant than the CTV baseline (second plot), they end up converging to an acceptable result.

The situation improves significantly as soon as the successive convexification loop is introduced and repeated. A progressive decrease of dispersions is visible in Fig. 7b ($N_{\mathrm{P}} = 1$) and Fig. 7c ($N_{\mathrm{P}} = 2$), and flight path constraints are successfully met by all the intermediate guidance solutions. In addition, for $N_{\mathrm{P}} = 2$, a propellant saving of 6.9% is



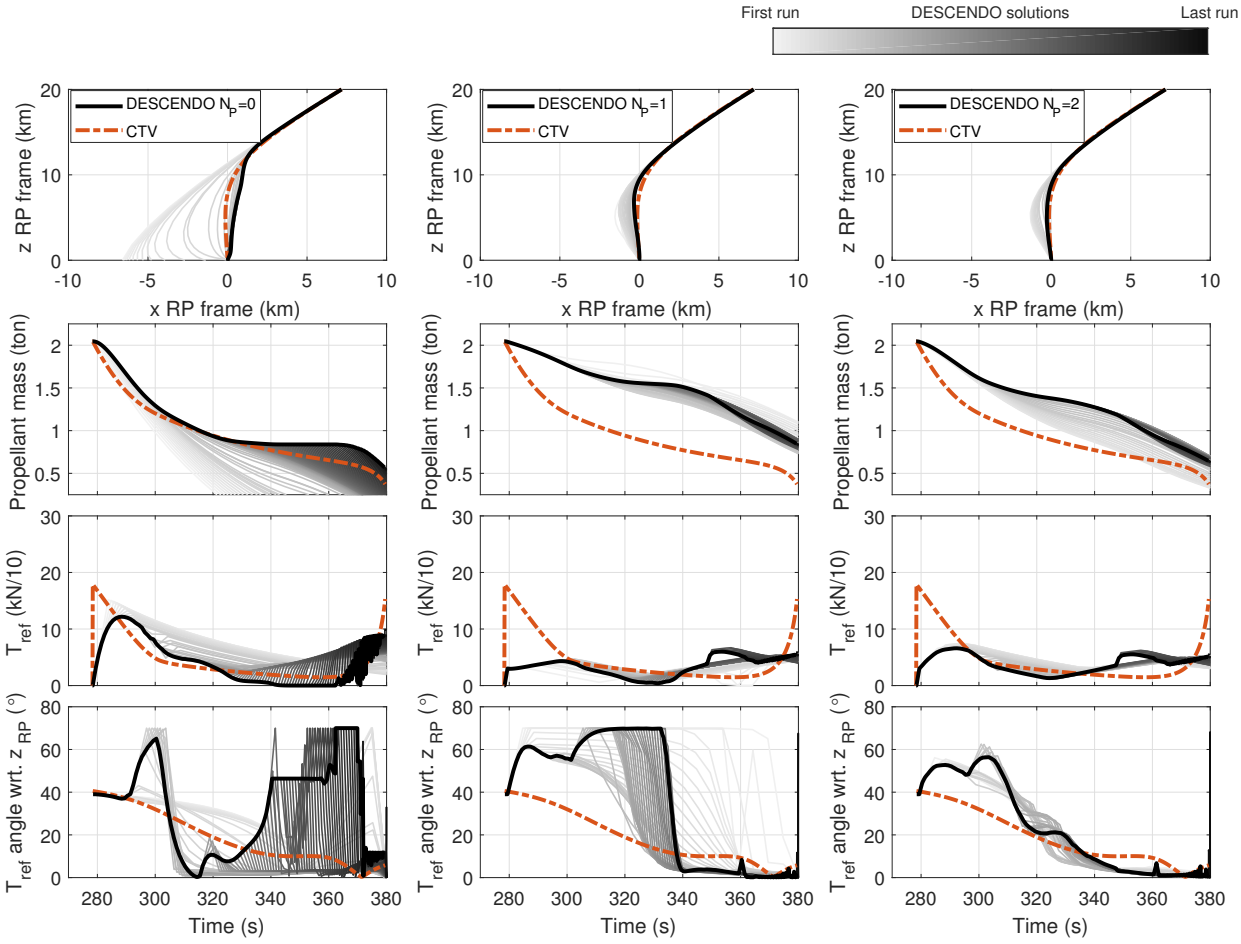
(a) $N_P = 0$ (b) $N_P = 1$ (c) $N_P = 2$

Fig. 7 DRL recovery using *DESCENDO*

achieved in comparison to the CTV baseline, and thrust magnitude and vertical direction profiles (the two bottom plots) become less abrupt (notice the time range between 310 and 350 seconds), while meeting the bounds of Table 2.

The exact same conclusions can be drawn for RTLS recovery, depicted in Fig. 8. Since the remaining propellant after the boostback burn is considerably lower than in the DRL case, the impact of the *DESCENDO* algorithm becomes clearer. Guidance solutions with $N_P = 0$, Fig. 8a, show larger dispersions (many of them requiring more propellant than what is actually available) and even degenerate into high control chattering close to touchdown (third plot). On the other hand, for $N_P = 2$ (Fig. 8c), a propellant saving of 16.4% is achieved in comparison to the CTV baseline. This value is not as high as for $N_P = 1$ (Fig. 8b) because it is achieved using a much smoother thrust profile, which once again indicates that a single indicator may not be enough for a fair D&L guidance comparison.

In order to have a better understanding of the convergence and computational efficiency of the algorithm, Fig. 9 illustrates the optimal value of the objective function $z[N] = \ln \hat{m}(t_f)$ (top plot of Fig. 9) and the cumulative ECOS runtime (bottom plot) as functions of the intermediate guidance solutions for the RTLS scenario with $N_P = \{0, 1, 2\}$.



(a) $N_P = 0$ (b) $N_P = 1$ (c) $N_P = 2$

Fig. 8 RTLS recovery using *DESCENDO*

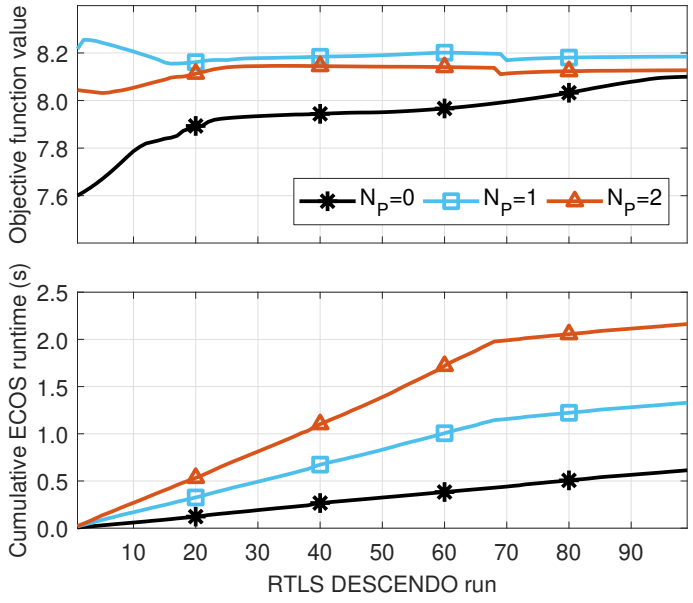


Fig. 9 Comparison of *DESCENDO* objective function and solver time during RTLS recovery

The top plot of Fig. 9 confirms that, although all the cases end up with similar objective function values, its accurate prediction takes place much later in the descent for $N_\mathrm{P} = 0$ (without successive convexification of aerodynamic effects). Solutions for $N_\mathrm{P} = 1$ and $N_\mathrm{P} = 2$ are relatively similar, which demonstrates the quick convergence of the algorithm, with $N_\mathrm{P} = 2$ slightly under $N_\mathrm{P} = 1$ as anticipated.

In terms of efficiency, the computational time naturally increases with N_P, but it only takes a total of 0.6, 1.3 or 2.2 seconds to compute the 100 seconds of recovery using $N_\mathrm{P} = \{0, 1, 2\}$, as evidenced in the bottom plot of Fig. 9. These values account for core solver time only and a small overhead needs to be added for problem parsing, but this is often negligible, especially if an embeddable solver such as FORCES [26] or CVXGEN [42] is employed.

It is interesting to notice that the execution speed increases significantly around the 70^{th} run for $N_\mathrm{P} = 1$ and $N_\mathrm{P} = 2$. This increase is due to the successive convexification switch-off for $h(t) \leq h_\mathrm{P}$ (where smaller velocities lead to less intense aerodynamic forces) and allows to reduce the overall computational time by about 30% with almost no optimality compromise (as seen in the top plot). Such a saving is not relevant for a real-world implementation since the system will have to be capable of coping with the successive convexification loops at higher altitudes, but it is very useful for preliminary mission feasibility studies where many simulations need to be executed.

In order to obtain a thorough understanding of the flight envelope encountered by the vehicle, detailed results for the same simulation of Fig. 8c are provided in Fig. 10. The phases of launch (from lift-off to separation), exo-atmospheric flight (from separation to recovery burn) and recovery (from recovery burn to touchdown) are distinguished in every plot using dash-dotted, dashed and continuous lines respectively.

The top-left plot of Fig. 10 shows the evolution of vertical and horizontal velocity as a function of altitude (in the vertical axis). During launch, velocity increases in both vertical and horizontal directions and separation occurs when the latter reaches its maximum value at an approximate altitude of 51 km. From that point, the vehicle continues to ascend until its vertical velocity becomes zero at an approximate altitude of 72 km and then plunges downwards due to the action of gravity.

In the meantime, the boostback burn results in the inversion of the horizontal component and reduction of its magnitude to approximately half of its value at separation. Then, at 25 km, recovery guidance is activated and the commanded burn brings both components to zero at the landing point. It is important to notice that horizontal velocity converges to this value significantly before their counterpart, which is critical for a vertical landing.

The bottom-left plot illustrates the evolution of dynamic pressure and heat flux as a function of the vehicle's Mach number. During launch, velocity increases and air density decreases, which causes Q and Q_H to tend to zero at lift-off and at maximum altitude (where $M \approx 5.3$) and to have a peak value in-between. These indicators then increase abruptly once the RLV starts to descend (at $M \approx 3.9$) and re-enters the atmosphere. At this point, recovery guidance is activated in order to manage the second peak value of these indicators and bring it to zero at the landing point.

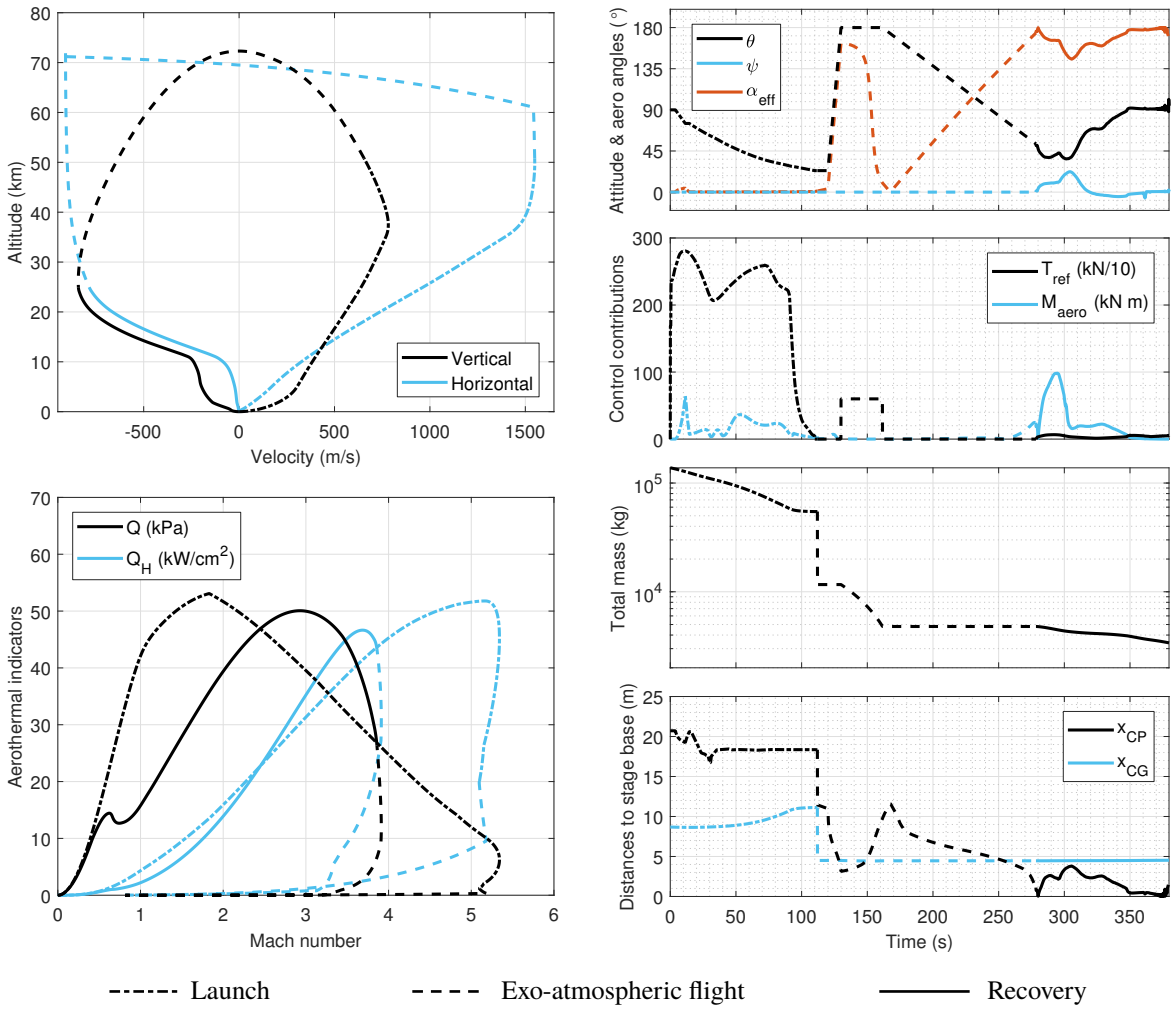


Fig. 10 RTLS flight mechanics results using *DESCENDO* and $N_P = 2$

The uppermost plot on the right-hand side of Fig. 10 shows the reference pitch and yaw angles $\{\theta_{\text{ref}}, \psi_{\text{ref}}\}$ as well as the total angle of attack α_{eff} over mission time. Lift-off, pitch over and gravity turn manoeuvres are clearly identified in the pre-programmed angles during launch and, in terms of angle of attack, during this phase it remains close to zero with a maximum value under 5 degrees around the pitch over phase. Subsequently, the reference pitch angle undergoes a rapid flip over manoeuvre, followed by a 30 seconds period with constant pitch during which the boostback burn takes place and by a second constant-rate manoeuvre prior to the recovery burn. The flip over manoeuvre causes the total angle of attack to follow the pitch variation, but the former angle returns to zero as soon as the horizontal velocity is inverted by the boostback burn. Finally, during the recovery, the reference attitude angles are computed by the guidance algorithm, which results in θ_{ref} converging to 90 degrees at touchdown, ψ_{ref} remaining close to zero (due to little aerodynamic couplings with the pitch motion) and α_{eff} close to 180 degrees.

The second right-hand plot illustrates the evolution of the thrust vector magnitude T_{ref} and aerodynamic moment $||\mathbf{M}_{\text{aero}}||$. Similar to the previous plot, T_{ref} is pre-programmed for launch and computed online by the guidance algorithm

during recovery. The reference associated with the boostback burn is also clearly visible. The aerodynamic moment to be compensated, as expected, is more demanding in zones of high dynamic pressure and angle of attack.

The third right-hand plot on the right-hand side of Fig. 10 shows the evolution of the vehicle's total mass using logarithmic scale for clarity. Here, four mass-depletion zones can be identified: launch burn, separation (sudden drop of m_{PL} at 112 seconds of flight), boostback and recovery burn. Finally, the bottom-right plot illustrates the longitudinal travel of CP and CG (relative to the booster's base) throughout the flight. While the former is governed by the aerodynamic environment encountered, the latter follows the depletion of mass. Hence, the same mass-depletion zones exist, although post-launch variations cannot be distinguished as changes in mass are comparatively small. In any case, the RLV is inherently unstable during powered flight since the CP is located in front of the CG during ascent and behind it during descent.

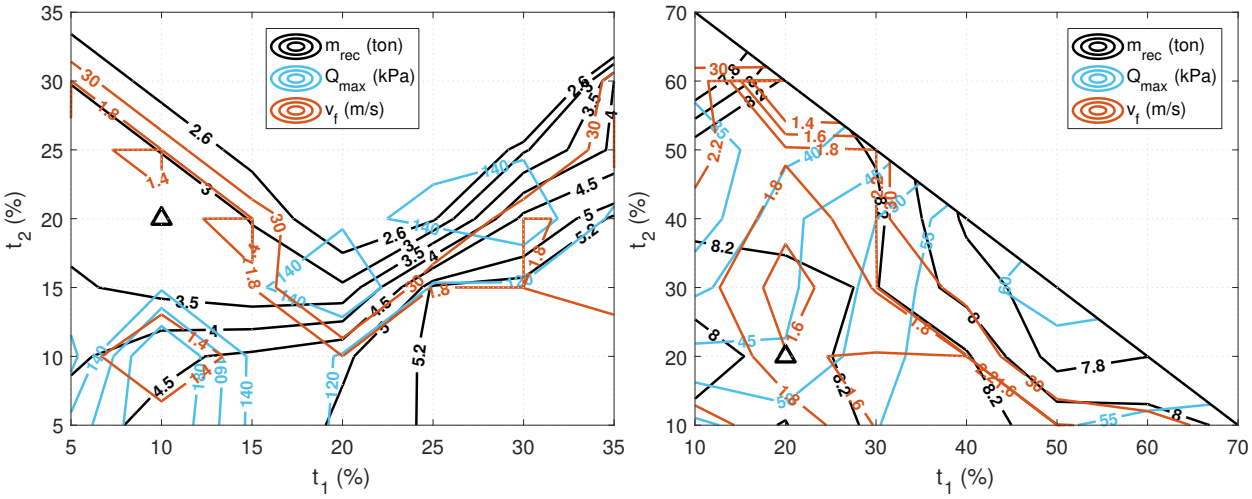
As an overview, the recovery metrics achieved with this approach are $m_{\mathrm{rec}} = 8.06$ ton, $Q_{\max} = 50.1$ kPa and $v_f = 1.63$ m/s. Comparing these results with those of the baseline CTV case (specifically, the choice set at the end of Sec. III.C, i.e. $m_{\mathrm{rec}} = 8.37$ ton, $Q_{\max} = 60$ kPa and $v_f = 2$ m/s), it is clear that the *DESCENDO* algorithm enables a significant improvement for all the indicators (most noticeably a simultaneous decrease of 310 kg propellant mass and 16% dynamic pressure), which was exactly the main motivation behind its development.

The same trade-off maps approach of Sec. III could be repeated to further improve these results by exploiting the parameter space of start altitude and final time. Alternatively, it is also interesting to utilise similar trade-off maps but to analyse the performance impact of explicitly splitting the recovery into two separate burns. To do so, Fig. 11a and 11b show DRL and RTLS maps for the parameter space of t_1 (relative duration of the first burn with respect to the total recovery phase) and t_2 (relative duration of the pause in-between burns). The single burn performance indicators (extracted from the simulation of Fig. 7c and 8c, respectively) are also provided in the captions for comparison.

For the DRL case (Fig. 11a), results are shown for t_1 and t_2 up to 35%. Although there is a large area of infeasible solutions (above the $v_f = 1.8$ m/s contour line, notice the abrupt transition to $v_f = 30$ m/s) as well as a few peaks of $Q_{\max}$ (most notably around $\{t_1, t_2\} = \{10, 10\}$ %), considerable performance gains can still be achieved. For instance, having a first burn of 10% followed by a pause of 20% (before the second burn) enables to bring m_{rec} and $Q_{\max}$ down from 5.05 to 3.13 ton and from 144 to 136 kPa, respectively, while only increasing v_f from 1.49 to 1.58 m/s.

Similar observations can be made for RTLS recovery (Fig. 11b), in which the parameter space $t_1 + t_2 \leq 80\%$ (i.e. allowing at least 20% of time for the second burn) was investigated. This map facilitates understanding the trade-off between propellant mass and dynamic pressure in the areas where v_f assumes feasible values (note that there is again a large region of infeasibility, with v_f above 30 m/s). As an example, keeping the same $v_f \approx 1.63$ m/s, $Q_{\max}$ can be reduced from 50.1 to 45.4 kPa by increasing m_{rec} from 8.06 to 8.10 ton while choosing $\{t_1, t_2\} = \{20, 20\}$ %.

To complement this result, the corresponding CTV and *DESCENDO* simulation using two burns is depicted



(a) DRL (with single burn $\{m_{\text{rec}},\ Q_{\max},\ v_f\} = \{5.05,\ 144,\ 1.49\}$)

(b) RTLS (with single burn $\{m_{\text{rec}},\ Q_{\max},\ v_f\} = \{8.06,\ 50.1,\ 1.63\}$)

Fig. 11 *DESCENDO* Trade-off maps for t_1 and t_2

(following the same lines of Fig. 8) in Fig. 12. This figure confirms the successful implementation of the *DESCENDO* algorithm, with the burn periods explicitly accounted for at all the intermediate guidance solutions (in opposition to the CTV baseline, in which the distinct burns are not enforced *a priori*). The pause is reflected in a constant propellant mass, zero thrust and undefined angle during the 20% of pause in-between burns (see the time range between 300 and 320 seconds in the bottom three plots).

VI. Conclusions

In reference [24], a preliminary performance assessment of reusable launcher downrange and RTLS recovery strategies was carried out using a baseline CTV guidance technique. The present article is a follow-up of that work, where room for performance improvement is exploited in two ways: first via a trade-off analysis of the physics of the powered descent and landing problem, and then by developing and implementing a more sophisticated guidance algorithm coined *DESCENDO* (Descending over Extended Envelopes using Successive Convexification-based Optimisation).

Regarding the trade-off analysis (Sec. III), improvements are possible because this approach allows to have a quick and clear mapping of the impact of different guidance parameters and of the compromises that can be achieved in terms of recovery propellant, dynamic pressure or touchdown speed. Nonetheless, major improvements do require the application of more sophisticated guidance techniques. The *DESCENDO* algorithm (Sec. IV) is a convex optimisation-based path planning strategy developed to meet this goal while providing a balance between computational efficiency and trajectory optimality that is suitable for the extended flight envelope encountered by reusable launchers. Using the reusable launcher benchmark of Sec. II, the effectiveness of *DESCENDO* and its superiority over the baseline CTV guidance were verified.

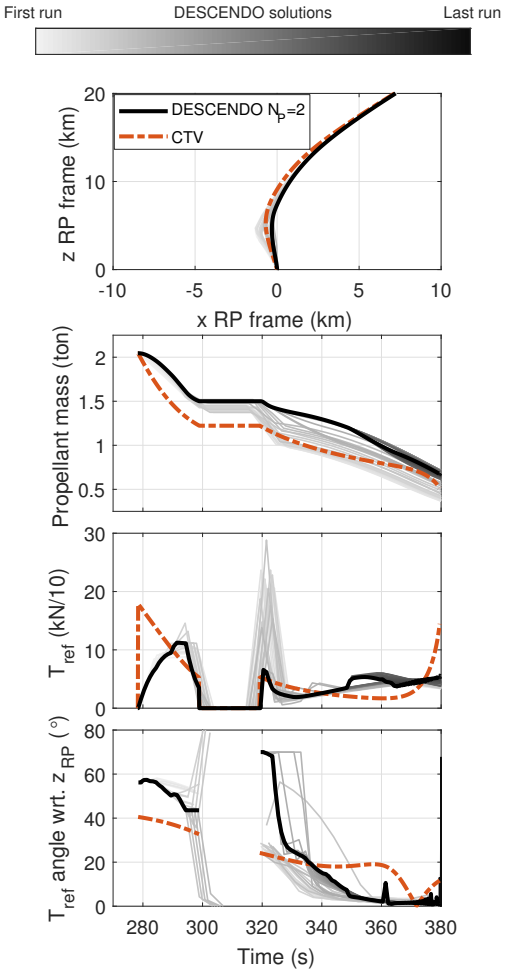


Fig. 12 RTLS recovery using two burns $\{t_1,\ t_2\} = \{20,\ 20\}$ %

But the major significance of the *DESCENDO* algorithm and of the trade-off map method lies in the fact that they enable a less conservative exploitation of the parameter trade-off space. The reduced conservatism, in turn, supports a deeper understanding of reusable flight mechanics, as well as guidance and control margin policies related to the physical limits of performance. Thanks to this understanding, the activity under which this study was carried out is continuing with the investigation of the impact of attitude control on aerothermal loads and of its minimisation using robust load prediction and relief algorithms.

Funding Sources

This work is funded by an ESA Networking Partnering Initiative (NPI) contract No. 4000119571/17/NL/MH with Dr. Stephan Theil (DLR-Bremen) as project coordinator. Mr. Simplício is also the recipient of a Doctoral Training Partnership (DTP) award by the UK EPSRC.

Acknowledgements

The authors would like to acknowledge and thank the support and help of Dr. Stephan Theil (DLR-Bremen, Germany) and Christophe Roux (AVIO, Italy).

References

[1] Klumpp, A., "Apollo lunar descent guidance," *Automatica*, Vol. 10, No. 2, 1974, pp. 133–146. doi:10.1016/0005-1098(74)90019-3.

[2] Li, S., and Jiang, X., "Review and prospect of guidance and control for Mars atmospheric entry," *Progress in Aerospace Sciences*, Vol. 69, 2014, pp. 40–57. doi:10.1016/j.paerosci.2014.04.001.

[3] Yoshimitsu, T., Kawaguchi, J., Hashimoto, T., Kubota, T., Uo, M., Morita, H., and Shirakawa, K., "Hayabusa – final autonomous descent and landing based on target marker tracking," *Acta Astronautica*, Vol. 65, 2009, pp. 657–665. doi:10.1016/j.actaastro.2009.01.074.

[4] Geurts, K., Fantinati, C., Ulamec, S., and Willnecker, R., "Rosetta Lander: On-Comet Operations Preparation and Planning," *The AIAA SpaceOps 2014 Conference*, Passadena, CA, 2014. doi:10.2514/6.2014-1752.

[5] Williams, B., Antreasian, P., Carranza, E., Jackman, C., Leonard, J., Nelson, D., Page, B., Stanbridge, D., Wibben, D., Williams, K., Moreau, M., Berry, K., Getzandanner, K., Liounis, A., Mashiku, A., Highsmith, D., Sutter, B., and Lauretta, D., "OSIRIS-REx Flight Dynamics and Navigation Design," *Space Science Reviews*, Vol. 214, No. 69, 2018. doi:10.1007/s11214-018-0501-x.

[6] Blackmore, L., "Autonomous Precision Landing of Space Rockets," *The Bridge on Frontiers of Engineering*, Vol. 4, No. 46, 2016, pp. 15–20.

[7] Starr, B., Yunis, I., and Aaron, O., "Use of Flexible Body Coupled Loads in Assessment of Day of Launch Flight Loads," *The 2011 AIAA Atmospheric Flight Mechanics Conference*, Portland, OR, 2011. doi:10.2514/6.2011-6465.

[8] Special Issue on Computational Guidance and Control, *Journal of Guidance, Control, and Dynamics*, Vol. 40, No. 2, 2017.

[9] Açikmeşe, B., and Ploen, S., "Convex Programming Approach to Powered Descent Guidance for Mars Landing," *Journal of Guidance, Control, and Dynamics*, Vol. 30, No. 5, 2007, pp. 1353–1366. doi:10.2514/1.27553.

[10] Blackmore, L., Açikmeşe, B., and Scharf, D., "Minimum-Landing-Error Powered-Descent Guidance for Mars Landing Using Convex Optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 33, No. 4, 2010, pp. 1161–1171. doi:10.2514/1.47202.

[11] Liu, X., and Lu, P., "Solving Nonconvex Optimal Control Problems by Convex Optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 37, No. 3, 2014, pp. 750–765. doi:10.2514/1.62110.

[12] Mao, Y., Szmuk, M., and Açikmeşe, B., "Successive Convexification of Non-Convex Optimal Control Problems and Its Convergence Properties," *The 55th IEEE Conference on Decision and Control*, Las Vegas, NV, 2016. doi:10.1109/CDC.2016.7798816.

[13] Açikmeşe, B., Aung, M., Casoliva, J., Mohan, S., Johnson, A., Scharf, D., Masten, D., Scotkin, A., Wolf, S., and Regehr, M., "Flight Testing of Trajectories Computed by G-FOLD: Fuel Optimal Large Divert Guidance Algorithm for Planetary Landing," *The 23rd AAS/AIAA Spacefilght Mechanics Meeting*, Kauai, HI, 2013.

[14] Scharf, D., Regehr, M., Vaughan, G., Benito, J., Ansari, H., Aung, M., Johnson, A., Casoliva, J., Mohan, S., Dueri, D., Açikmeşe, B., Masten, D., and Nietfeld, S., "ADAPT Demonstrations of Onboard Large-Divert Guidance with a VTVL Rocket," *The 2014 IEEE Aerospace Conference*, Passadena, CA, 2014. doi:10.1109/AERO.2014.6836462.

[15] Dueri, D., Açikmeşe, B., Scharf, D., and Harris, M., "Customized Real-Time Interior-Point Methods for Onboard Powered-Descent Guidance," *Journal of Guidance, Control, and Dynamics*, Vol. 40, No. 2, 2017, pp. 197–212. doi:10.2514/1.G001480.

[16] Scharf, D., Açikmeşe, B., Dueri, D., Benito, J., and Casoliva, J., "Implementation and Experimental Demonstration of Onboard Powered-Descent Guidance," *Journal of Guidance, Control, and Dynamics*, Vol. 40, No. 2, 2017, pp. 213–229. doi:10.2514/1.G000399.

[17] Ebrahimi, B., Bahrami, M., and Roshanian, J., "Optimal sliding-mode guidance with terminal velocity constraint for fixed-interval propulsive maneuvers," *Acta Astronautica*, Vol. 62, 2008, pp. 556–562. doi:10.1016/j.actaastro.2008.02.002.

[18] Hawkins, M., Guo, Y., and Wie, B., "Spacecraft Guidance Algorithms for Asteroid Intercept and Rendezvous Missions," *International Journal of Aeronautical and Space Sciences*, Vol. 13, No. 2, 2012, pp. 154–169. doi:10.5139/IJASS.2012.13.2.154.

[19] Guo, Y., Hawkins, M., and Wie, B., "Applications of Generalized Zero-Effort-Miss/Zero-Effort-Velocity Feedback Guidance Algorithm," *Journal of Guidance, Control, and Dynamics*, Vol. 36, No. 3, 2013, pp. 810–820. doi:10.2514/1.58099.

[20] Joffre, E., Zamaro, M., Silva, N., Marcos, A., and Simplício, P., "Trajectory design and guidance for landing on Phobos," *Acta Astronautica*, Vol. 151, 2018, pp. 389–400. doi:10.1016/j.actaastro.2018.06.024.

[21] Liu, X., Shen, Z., and Lu, P., "Entry Trajectory Optimization by Second-Order Cone Programming," *Journal of Guidance, Control, and Dynamics*, Vol. 39, No. 2, 2016, pp. 227–241. doi:10.2514/1.G001210.

[22] Liu, X., "Fuel-Optimal Rocket Landing with Aerodynamic Controls," *The AIAA SciTech 2017 Forum*, Grapevine, TX, 2017. doi:10.2514/6.2017-1732.

[23] Wang, Z., and Grant, M., "Constrained Trajectory Optimization for Planetary Entry via Sequential Convex Programming," *Journal of Guidance, Control, and Dynamics*, Vol. 40, No. 10, 2017, pp. 2603–2615. doi:10.2514/1.G002150.

[24] Simplício, P., Marcos, A., and Bennani, S., "Reusable Launchers: Development of a Coupled Flight Mechanics, Guidance and Control Benchmark," *submitted to the Journal of Spacecraft and Rockets*, 2018.

[25] Szmuk, M., Açikmeşe, B., Berning Jr., A., and Huntington, G., "Successive Convexification for Fuel-Optimal Powered Landing with Aerodynamic Drag and Non-Convex Constraints," *The AIAA SciTech 2016 Forum*, San Diego, CA, 2016. doi:10.2514/6.2016-0378.

[26] Jerez, J., Merkli, S., Bennani, S., and Strauch, "FORCES-RTTO: A Tool for On-board Real-time Autonomous Trajectory Planning," *The 10th International ESA Conference on Guidance, Navigation and Control Systems*, Salzburg, Austria, 2017.

[27] Greensite, A. L., *Analysis and Design of Space Vehicle Flight Control Systems*, Vol. II and VII, Spartan, 1970.

[28] Pavlis, N., Holmes, S., Kenyon, S., and Factor, J., "An Earth Gravitational Model to Degree 2160: EGM2008," *The 2008 General Assembly of the European Geosciences Union*, Vienna, Austria, 2008.

[29] *U.S. Standard Atmosphere*, NASA Technical Reports Server, 1976.

[30] Bianchi, S., "VEGA, the European small launcher: Development status, future perspectives, and applications," *Acta Astronautica*, Vol. 63, 2008, pp. 416–427. doi:10.1016/j.actaastro.2007.12.058.

[31] Zarchan, P., *Tactical and Strategic Missile Guidance*, 2nd ed., AIAA Progress in Astronautics and Aeronautics, 1994. doi:10.2514/4.868948.

[32] Simplício, P., Marcos, A., Joffre, E., Zamaro, M., and Silva, N., "Systematic Performance-oriented Guidance Tuning for Descent & Landing on Small Planetary Bodies," *Acta Astronautica, Special Issue on EUCASS 2017 System & GNC, in press*, 2018. doi:10.1016/j.actaastro.2018.07.028.

[33] Pascucci, A., Bennani, S., and Bemporad, A., "Model Predictive Control for Powered Descent Guidance and Control," *The 2015 European Control Conference*, Linz, Austria, 2015. doi:10.1109/ECC.2015.7330732.

[34] Reynolds, T., and Mesbahi, M., "Small Body Precision Landing via Convex Model Predictive Control," *The AIAA SPACE and Astronautics Forum and Exposition*, Orlando, FL, 2017. doi:10.2514/6.2017-5179.

[35] Eren, U., Prach, A., Koçer, B., Raković, S., and Açikmeşe, B., "Model Predictive Control in Aerospace Systems: Current State and Opportunities," *Journal of Guidance, Control, and Dynamics*, Vol. 40, No. 7, 2017. doi:10.2514/1.G002507.

[36] Nesterov, Y., and Nemirovski, A., *Interior-Point Polynomial Algorithms in Convex Programming*, 1st ed., SIAM, 1994. doi:10.1137/1.9781611970791.

[37] Sturm, J., "Implementation of Interior Point Methods for Mixed Semidefinite and Second Order Cone Optimization Problems," *Optimization Methods and Software*, Vol. 17, No. 6, 2002, pp. 1105–1154. doi:10.1080/1055678021000045123.

[38] Szmuk, M., and Açikmeşe, B., "Successive Convexification for 6-DoF Mars Rocket Powered Landing with Free-Final-Time," *The AIAA SciTech 2018 Forum*, Kissimmee, FL, 2018. doi:10.2514/6.2018-0617.

[39] Lee, U., and Mesbahi, M., "Constrained Autonomous Precision Landing via Dual Quaternions and Model Predictive Control," *Journal of Guidance, Control, and Dynamics*, Vol. 40, No. 2, 2017, pp. 292–308. doi:10.2514/1.G001879.

[40] Grant, M., and Boyd, S., "CVX: Matlab Software for Disciplined Convex Programming, version 2.1," Online available at `http://cvxr.com/cvx`, 2014.

[41] Domahidi, A., Chu, E., and Boyd, S., "ECOS: An SOCP solver for embedded systems," *The 2013 European Control Conference*, Zurich, Switzerland, 2013. doi:10.23919/ECC.2013.6669541.

[42] Mattingley, J., and Boyd, S., "CVXGEN: a code generator for embedded convex optimization," *Optimization and Engineering*, Vol. 13, No. 1, 2012. doi:10.1007/s11081-011-9176-9.